

JAN 23 2009

Appl. No. 10/798,459  
Declaration of prior invention  
Reply to office action of October 28, 2008

PATENT  
Case No. N0184US

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

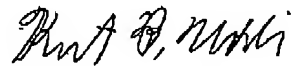
Appl. No. : 10/798,459  
Applicants : Kurt Brooks Uhler, et al.  
Filed : March 11, 2004  
Titled : METHOD AND SYSTEM FOR USING GEOGRAPHIC DATA  
IN COMPUTER GAME DEVELOPMENT

**DECLARATION UNDER 37 C.F.R. § 1.131**

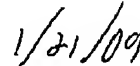
The undersigned, KURT BROOKS UHLIR, CHRISTOPHER DOUGHERTY,  
MICHAEL V. SHUMAN, and ROY CASINO, hereby declare that:

1. We are co-inventors of the invention described and claimed in the above-identified patent application.
2. Before May 21, 2003, we invented a new method for deriving products from a source geographic database. Part of this new method included using or extracting data from a source geographic database to provide a dataset used in developing video games that depicted real geographic locales as part of play scenarios. An application programming interface used for developing the video games from the specific dataset was also provided. The source geographic database included navigation attributes including data representing road segments, and the source geographic database was also used to provide a dataset to develop real-world navigation systems.
3. Before May 21, 2003, we prepared invention disclosure statements describing our inventive ideas. We provided the invention disclosure statements to the Legal Department of the assignee of the subject patent application. A copy of an invention disclosure statement prepared by us prior to May 21, 2003 fully disclosing the claimed invention is attached hereto (Exhibit 1).
4. All statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or

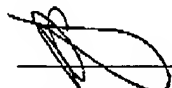
imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful statements may jeopardize the validity of the application or any patent issuing thereon.



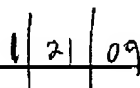
KURT BROOKS UHLIR



Date



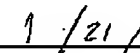
CHRISTOPHER DOUGHERTY



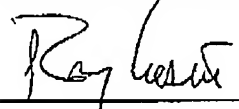
Date



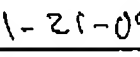
MICHAEL V. SHUMAN



Date



ROY CASINO



Date

NAVTEQ North America, LLC  
425 West Randolph Street  
Chicago, IL 60606  
(312) 780-3054



JAN 23 2009

The Game API will offer functions broken into four key areas that differ from current NTC APIs:

- I. Functions to convert NAVTECH data to other formats suitable for games
- II. Functions for high-speed display necessary for games
- III. Functions for aesthetically pleasing display formats necessary for games
- IV. Function to easily integrate NAVTECH data with other data content necessary for games

Each of the four areas will be described in more detail below.

- I. Functions to convert NAVTECH data to other formats suitable for games

**Overview:**

These functions allow conversion of NAVTECH data to formats for use with 3-d display, cell phone display, PC gaming via internet display, and others. The idea here is that the NAVTECH database should remain focused on navigation applications. In turn, these functions will use the NAVTECH database, but offer data necessary for game play in one of the game formats. For example, a "cannonball run" type of racing game would require the NAVTECH data to appear in 3-d format as if a car were driving the roads. This would allow drivers to race across the United States (or other region), allow them to choose to drive on any road, allow the games to drive realistic routes with real knowledge of speed limits, signage, and more integrated into their game play.

**Example:**

3-d display: The 3-d display functions will allow games to "travel" along roads from a driver's perspective. For example, a driving game could allow one to race a route from Chicago to St. Louis.

**Functions:**

*3\_d\_display(link ID, location, direction\_of\_travel, DisplayInfo)*

link ID (input)...the NAVTECH link ID of the desired link for display

location (input)...the actual location along the link (expressed in percentage along the link)

direction\_of\_travel (input)...direction of travel (from reference node or to reference node)

DisplayInfo (output)...a structure (not defined here) that holds all information necessary for displaying a 3-d image of the road and related attributes. The image drawn on a display using this information would be a computer rendered view of what a driver would see from a car located on the road at this point.

**Example:**

internet\_display: The internet display functions will allow incremental updates of internet displays for game play. For example, a first display screen would show full detail of a certain geographic area in 2d or full 3d. Further updates to the screen will only send necessary information to indicate changes to the current display. This method of sending only changes will allow faster updates necessary for game play over the internet. A function like this could allow a set of gamers across the nation connected only by internet connections to play competitive games based on NAVTECH data with each other in real-time.

*internet\_display(region, 2d\_3d\_flag, location, DisplayInfo)*

region (input)...bounding box region to display

2d\_3d\_flag (input)...whether 2d or 3d display should be shown

location (input)... gamer's location on the requested map

DisplayInfo (output)... a structure (not defined here) that holds all information necessary for displaying an image of the road and related attributes. Further updates to this display would only send changes to allow for fast transmission across internet connections and fast display updates.

**More:**

CONFIDENTIAL

Exhibit 1 (2/4)

Those are only two representative type functions that could exist in a game API. Clearly, numerous additional functions will be necessary in the actual implementation.

II. Functions for high-speed display necessary for games

Overview:

Current NavTools display tools are not optimized for high-speed updates or continuous movement. The game API will require functions that support these types of scenarios. This will allow a driving game, for example, to give the appearance that a vehicle is being driven along a road way created using NAVTECH data.

Requirements:

No new functions will necessarily be required versus current NavTools functions as much as more optimization will be required. Concepts such as high speed polygon display and other video game derived display technologies will have to be incorporated into the game API tool set. This integration of video game technology into a game API will allow the NAVTECH database to be drawn in 2d or 3d at very high speeds.

III. Functions for aesthetically pleasing display formats necessary for games

Overview:

Current NTC APIs display maps for navigation purposes. The game API will require changes so that displays can be oriented to more aesthetically pleasing display formats. Formats that include road widths and lane stripes will be incorporated into the display functions. Furthermore, attributes such as road names will be optional in the display as road names may not always be necessary for games. Moreover, games may choose to leave off the road names for the sake of speed.

Requirements:

The display formats for games will require that NAVTECH data (intended for navigation applications) will have additional visual information attached during the process of preparing for display. For example, a game may query for a display that includes stripes on the lanes and shoulders along the edges of the road. Further examples are the inclusion of road signs so that a game player sees a speed limit sign, for example, pass by as he/she drives.

Function:

*set\_display\_parameters(visual\_attribute\_list)*

*visual\_attribute\_list* (input)...this parameter allows one to set visual attributes such as whether lane stripes should be added to the roads for display, whether signs should be added to the display, whether shoulders or curbs should be added, and other attributes of this sort intended to visually aid a driver playing a game and not necessarily aiding in navigation.

IV. Function to easily integrate NAVTECH data with other data content necessary for games

Overview:

Many games will also desire to integrate information such as the look of other vehicles on the roads, traffic information on the roads, the look of buildings along the road, and other such examples. The game API will allow the game designers access to the polygonal display tools so that these additional features could easily be integrated with the NAVTECH data. For example, a game designer may want to populate the roads being driven with actual vehicles on the market. In that case, the game designer would obtain vehicle modeling information from other sources and feed it to the game API so that the vehicles could be placed on a 3-d NAVTECH road map display.

Function:

CONFIDENTIAL

Exhibit 1 (3/4)

*integrate\_external\_information(polygonal\_information, location)*

*polygonal\_information* (input)...the polygon modeling information necessary to render a particular geometrical shape such as a building or vehicle.

*location* (input)...the location to display the *polygonal\_information*. This would allow one to place buildings along the road or other vehicles along the road.

Please place an "X" next to the appropriate statement:

X No design documents exist

The following design documents exist (and copies are attached): \_\_\_\_\_

Signature: \_\_\_\_\_  
(of preparer-developer)

Date: \_\_\_\_\_

Type Name: Roy Casino

Signature(s) of Contributing Developers:

1. Name: _____	Date: _____
2. Name: _____	Date: _____
3. Name: _____	Date: _____
4. Name: _____	Date: _____
5. Name: _____	Date: _____
6. Name: _____	Date: _____
7. Name: _____	Date: _____

CONFIDENTIAL

Exhibit 1 (4/4)